

# Greedy Oriented Flows

Ulrich Faigle<sup>1</sup> · Walter Kern<sup>2</sup> · Britta Peis<sup>3</sup>

Received: 18 September 2014 / Accepted: 23 March 2017  
 © The Author(s) 2017. This article is an open access publication

**Abstract** We investigate the following greedy approach to attack linear programs of type  $\max\{1^T x \mid l \leq Ax \leq u\}$  where  $A$  has entries in  $\{-1, 0, 1\}$ : The greedy algorithm starts with a feasible solution  $x$  and, iteratively, chooses an improving variable and raises it until some constraint becomes tight. In the special case, where  $A$  is the edge-path incidence matrix of some digraph  $G = (V, E)$ , and  $l = 0$ , this greedy algorithm corresponds to the Ford–Fulkerson algorithm to solve the MAX  $(s, t)$ -FLOW PROBLEM in  $G$  w.r.t. edge-capacities  $u$ . It is well-known that the Ford–Fulkerson algorithm always terminates with an optimal flow, and that the number of augmentations strongly depends on the choice of paths in each iteration. The Edmonds–Karp rule that prefers paths with fewer arcs leads to a running time of at most  $|E|^2$  augmentations. The paper investigates general types of matrices  $A$  and preference rules on the variables that make the greedy algorithm efficient. In this paper, we identify conditions that guarantee for the greedy algorithm not to cycle, and/or optimality of the greedy algorithm, and/or to yield a quadratic (in the number of rows) number of augmentations. We illustrate our approach with flow and circulation problems on regular oriented matroids.

**Keywords** Greedy algorithm · Regular matroid · Max flow · Cycle cancelling

---

✉ Walter Kern  
 w.kern@math.utwente.nl

Ulrich Faigle  
 faigle@zpr.uni-koeln.de

Britta Peis  
 britta.peis@oms.rwth-aachen.de

<sup>1</sup> Mathematisches Institut, Universität zu Köln, Weyertal 80, 50931 Köln, Germany

<sup>2</sup> Universiteit Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

<sup>3</sup> RWTH Aachen, Kackertstraße 7, 52072 Aachen, Germany

## 1 Introduction

Consider a linear program of the form

$$\max \left\{ \mathbf{1}^T x \mid l \leq Ax \leq u \right\}, \quad (1)$$

where  $A$  is a  $\{-1, 0, 1\}^{m \times n}$ -matrix whose row and column index sets we denote by  $E$  and  $\mathcal{P}$ , respectively, and  $l, u \in \mathbb{R}^m$  are lower and upper capacities (i.e.,  $|E| = m$  and  $|\mathcal{P}| = n$ ). To simplify notation, let us identify each column index  $P \in \mathcal{P}$  with the corresponding matrix column  $P \in \{-1, 0, 1\}^m$ , and each row index  $e \in E$  with the corresponding constraint. Hence,  $P$  is a vector with components  $P_e \in \{-1, 0, 1\}$  for  $e \in E$ .

Our investigation is motivated by the classical max-flow problem of Ford and Fulkerson [16] relative to the directed graph  $G = (V, E)$  with designated *source*  $s \in V$  and *sink*  $t \in V$  and capacity restrictions  $u \in \mathbb{R}_+^{|E|}$ . Let  $\mathcal{P}$  denote the collection of all (characteristic vectors of)  $(s, t)$ -paths  $P$ , in the sense that  $P_e = 1$  [ $P_e = -1$ ] if  $e$  is traversed by  $P$  in forward [backward] direction, and  $P_e = 0$  if  $e$  is not traversed by  $P$  at all. Indeed, choosing  $l = 0$ , problem (1) becomes the Ford–Fulkerson problem

$$\max \left\{ \mathbf{1}^T x \mid 0 \leq Ax \leq u \right\}. \quad (2)$$

A natural approach to solve (2) is to start with the all-zero vector  $x = 0$ , and to iteratively increase one of the variables (among those that can be increased at all) as far as possible, i.e., until some constraint  $e \in E$  becomes tight w.r.t. either the lower or the upper capacity bound. Ford and Fulkerson suggest the following iterative algorithmic solution procedure for (2):

Start with a feasible solution  $x_0$  (for example,  $x_0 = 0$ ). Where  $x$  is the current solution, consider the path collection

$$\mathcal{P}^x := \{P \in \mathcal{P} \mid \exists \varepsilon > 0 \text{ with } 0 \leq Ax + \varepsilon P \leq u\}. \quad (3)$$

The elements in  $\mathcal{P}^x$  are called *residual* or *augmenting* paths w.r.t.  $x$ . Unless  $\mathcal{P}^x = \emptyset$ , pick some augmenting path  $P \in \mathcal{P}^x$  and raise the variable  $x_P$  as much as is feasibly permitted in order to achieve a new solution  $x$ . If the algorithm terminates after a finite number of steps with  $x^*$ , it follows from the max-flow min-cut theorem that an optimal feasible flow  $f^* = \sum x_P^* P$  has been found. Furthermore, it is clear that  $x^*$  is integral when the capacity bound  $u$  is integral.

The Ford–Fulkerson algorithm may not be finite if augmenting paths are chosen arbitrarily. However, if the augmenting path is always selected according to the *Edmonds–Karp rule* ([10]), i.e., as a path  $P \in \mathcal{P}^x$  with a minimal number of arcs, termination after at most  $|V| \cdot |E|$  augmentations is guaranteed (see, e.g., [1] or [12]). So we find that the running time of the Ford–Fulkerson algorithm depends strongly on our preferences for breaking possible ties among augmenting paths.

*The General Oriented Flow Problem* We now turn to the general max-flow problem (1) where we consider arbitrary  $\{-1, 0, 1\}$ -matrices (instead of just edge-path incidence matrices), and call it the *general oriented flow problem*. In analogy with the Ford–Fulkerson setting, we refer to the members of  $\mathcal{P} \subseteq \{-1, 0, 1\}^m$  as (*generalized oriented*) *paths*. We approach the general oriented flow problem (1) algorithmically in the spirit of Ford–Fulkerson and start with an initial feasible solution  $x_0$ . Where  $\mathcal{P}^x$  is defined as in (3), we iteratively try to improve the current feasible solutions  $x$  by picking an *augmenting* residual generalized path  $P \in \mathcal{P}^x$  and taking  $x + \varepsilon_P P$  as our new current solution with  $\varepsilon_P$  as large as possible subject to the given capacity constraints.

We assume a initial feasible solution  $x_0$  to be given. In the case  $l \leq 0 \leq u$ , for example,  $x_0 = 0$  will do. Moreover, if  $l, u$  and  $x_0$  are integral, this algorithm returns an integral solution  $x^*$  (if it terminates at all) for the following reason: in each iteration the algorithm increases the current integral solution  $x$  by raising one variable  $x_P$  by an integral amount (namely, the capacity of an element, say  $\hat{e}$ , which becomes tight, minus the current flow  $\sum_{P \in \mathcal{P}: \hat{e} \in P^+} x_P - \sum_{P \in \mathcal{P}: \hat{e} \in P^-} x_P$ ).

*Ternary Matrices and Families of Signed Sets* Note that any ternary matrix  $A \in \{-1, 0, 1\}^{|E| \times |\mathcal{P}|}$  encodes a family  $\mathcal{P}$  of oriented (a.k.a. *signed*) subsets of the set  $E$ , where an *oriented* subset of  $E$  is a pair  $P = (P^+, P^-)$  of disjoint subsets of  $E$ . That is, the elements in  $P^+$  [ $P^-$ ] correspond to the positive [negative] entries of column  $P$ . Let  $|P| := |P^+| + |P^-|$  denote the *length* of the oriented set  $P = (P^+, P^-)$ . For later use we note that the column ordering of the matrix  $A$  even defines a total order of the associated family of signed sets.

*Performance on Ternary Matrices in General* In contrast to the classical max-flow problem, however, a solution  $x^*$  returned by the algorithm above is not guaranteed to be optimal. Consider, for example, the general oriented flow problem with capacity constraints

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} x \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

If we augment the first variable (i.e., choose the first path for augmentation), the algorithm terminates with objective value 1 while an optimal solution is achievable by raising the second and third variable up to 1 instead.

Even when the algorithm finds an optimal solution, its running time performance might be extremely poor. Consider, i.e., the classical textbook example for exponential time flow augmentation: A large natural number  $M$  and the constraints

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} x \leq \begin{bmatrix} M \\ M \\ 1 \\ M \\ M \end{bmatrix}.$$

If we always augment along the residual path (variable) with the smallest column index, the algorithm jumps  $M$  times back and forth between the first and second column, each time raising the current variable by only 1. If preference were given to the third and fourth column, however, the algorithm would terminate after only two iterations with an optimal solution.

Thus, even when the optimality of the algorithm is guaranteed, regardless of the preference relation with respect to which the paths are chosen for augmentation, the choice of the preference relation is crucial when it comes to runtime analysis. Consider, for example, the ordinary max flow setting: We have already remarked that the Edmonds–Karp rule to prefer paths with fewer arcs bounds the number of augmentations in a classical max-flow problem on  $G = (V, E)$  by  $|V| \cdot |E|$ . In the special case of a plane graph  $G$ , Borradaile and Klein [6] showed that, at least after some preprocessing, only  $O(|E| \log |E|)$  iterations suffice when preference is given to the respective “leftmost” paths in the residual path structures  $\mathcal{P}^x$  (see also Sect. 3.1 below).

**Greedy Algorithms** In order to formalize preference rules for the selection of augmenting paths  $P \in \mathcal{P}^x$ , we assume to be given an acyclic and transitive *preference relation*  $(\mathcal{P}, <)$  on the paths. For any feasible solution  $x$  for (1), we denote the collection of  $<$ -minimal augmenting paths by

$$\mathcal{P}_{<}^x := \{P \in \mathcal{P}^x \mid Q \not< P \ \forall Q \in \mathcal{P}^x\}.$$

The associated *greedy algorithm* is now the iterative algorithm of Ford–Fulkerson type with the augmentation rule

- (G) The current feasible solution  $x$  may be augmented along any path  $P \in \mathcal{P}_{<}^x$ . If no such  $P$  exists, the algorithm terminates and outputs  $x$ .

In the special case of the *trivial preference relation* (i.e.,  $P \not< Q$  for all  $P, Q \in \mathcal{P}$ ), we have  $\mathcal{P}^x = \mathcal{P}_{<}^x$ . So the greedy algorithm may choose any  $P \in \mathcal{P}^x$  for augmentation.

**Optimality** There is generally no guarantee that a solution returned by the greedy algorithm is optimal unless the linear program (1) has special structural properties. Let us say that (1) is *augmentation optimal* if the following is true:

- (O) A feasible solution  $x$  is optimal if  $\mathcal{P}^x = \emptyset$ .

**Assumptions** We assume throughout that  $|E|$  is finite and rather small, whereas  $|\mathcal{P}|$  might be big, i.e., exponential in  $|E|$ . In particular, we assume that the matrix  $A$  (resp. the signed system  $\mathcal{P}$ ), is not given explicitly, but *via* an oracle returning the next variable to be raised. Note that this assumption is justified by several examples where the matrix is not given explicitly (as this would be highly inefficient), but the next variable to be raised can be found efficiently. So the running time is essentially determined by the number of augmentations.

*Questions* Some immediate questions arise in the above context:

- Which structures  $\mathcal{P}$  guarantee the greedy solution to be optimal (and found after finitely many steps)—no matter what capacities are considered, and independent of the chosen preference relation  $<$ ?
- Which structures  $(\mathcal{P}, <)$  guarantee the number of greedy augmentations to be polynomially bounded in  $|E|$ ?
- Which structures  $(\mathcal{P}, <)$  guarantee the greedy algorithm to never raise a variable more than once?

*Our Results* We start by introducing *conformal systems* as a far reaching generalization of ordinary path systems in the subsequent Sect. 2. Additionally, we define the notion of *compatibility* as a condition on preference orders  $(\mathcal{P}, <)$  on conformal systems. In Sect. 3, we show that the compatibility condition on conformal systems  $(\mathcal{P}, <)$  ensures the greedy algorithm to raise none of the variables twice (Theorem 1) and to terminate after  $<g> \cdot |E|$  iterations, where  $<g>$  denotes the number of different values of a potential function  $g : \mathcal{P} \rightarrow \mathbb{R}$  on  $(\mathcal{P}, <)$  like, e.g., the *height* of  $(\mathcal{P}, <)$ . Conformal systems  $\mathcal{P}$  ordered by non-decreasing support-size are compatible, for example, and so is the lexicographic order of  $\mathcal{P}$  induced by a total ordering of  $|E|$ , or the left–right order of ordinary paths used by Borradaile and Klein for planar max flow problems. As one of the consequences, it follows that the greedy algorithm takes at most  $|E|^2$  iterations (Corollary 1) whenever  $(\mathcal{P}, <)$  is a conformal system that is ordered by non-decreasing support size, which generalizes the Edmonds–Karp variant of the classical Ford–Fulkerson algorithm.

Finally, in Sect. 4, we apply our results to the case of a path system corresponding to the system of circuits of a regular oriented matroid. It turns out (Theorem 3), that in this case a solution returned by the greedy algorithm is guaranteed to be optimal, independent of the capacities  $l, u \in \mathbb{R}^E$ , and independent of the preference order  $<$ . Since regular oriented matroids yield conformal systems, it follows that ordering such a system in a compatible way guarantees that none of the variables is raised twice. Moreover, it follows that the greedy algorithm on regular oriented matroids terminates already after at most  $|E|^2$  augmentations with an optimal solution whenever preference is given to paths of smaller support size.

The min-cost circulation problem has been studied on regular oriented matroids by Karzanov and McCormick [25]. We outline how our approach allows a straightforward primal analysis of the the generalized cycle canceling algorithm.

*Related Results* There is previous work on greedy solvability of linear programs. For example, [11] identifies a class of non-negative real constraint matrices that allow greedy optimization relative to any objective, right hand side and arbitrary box constraints. Also other work in this area typically focusses on  $(0, 1)$ -matrices (see, e.g., [13, 15, 17, 23], or [14]). We also mention the research on binary and ternary systems originated by Fujishige (cf. [19–21]). Already Bland [5] reported that the Edmonds–Karp variant generalizes to regular matroids. Unfortunately, a proof of this important result has never been published until now.

## 2 Conformal Systems and Compatible Orderings

We call two vectors  $P, Q \in \{-1, 0, 1\}^{|E|}$  *conforming* if

$$P^+ \cap Q^- = \emptyset = P^- \cap Q^+.$$

For example, if system  $\mathcal{P} \subseteq \{-1, 0, 1\}^{|E|}$  encodes the  $s - t$ -paths in a directed graph  $G = (V, E)$ , two paths  $P$  and  $Q$  are conforming whenever they do not traverse any edge in opposite direction.

**Definition 1** System  $\mathcal{P} \subseteq \{-1, 0, 1\}^{|E|}$  is said to be *conformal* if for all non-conforming elements  $P, Q \in \mathcal{P}$  there exist two designated elements in  $\mathcal{P}$ , call them  $P \wedge Q$  and  $P \vee Q \in \mathcal{P}$ , such that  $P \wedge Q$  and  $P \vee Q$  are conforming and

$$P \wedge Q + P \vee Q \leq P + Q, \quad (4)$$

where, for any two vectors  $x, y \in \mathbb{R}^E$ , we write  $x \leq y$  if  $x^+ \leq y^+$  and  $x^- \leq y^-$ .

Here, as usual,  $x^+$  and  $x^-$  are the two non-negative vectors of disjoint support such that  $x = x^+ - x^-$  (note that each  $x \in \mathbb{R}^{|E|}$  uniquely decomposes in two such vectors  $x^+$  and  $x^-$ ).

Throughout, we call  $P \wedge Q$  the *meet*, and  $P \vee Q$  the *join* of  $P$  and  $Q$ , although our definition does not correspond to the typical meet- and join-operation in lattices. Intuitively, a conformal system  $\mathcal{P}$  ensures a certain *uncrossing property*: whenever  $x$  is a feasible solution to (2) with  $\delta = \min\{x_P, x_Q\} > 0$ , subtracting  $\delta$  from  $x_P$  and  $x_Q$ , and adding  $\delta$  to  $x_{P \wedge Q}$  and  $x_{P \vee Q}$  yields a feasible solution as well.

### 2.1 Examples of Conformal Systems

It is not hard to see that (characteristic vectors of) paths in a given directed graph  $G = (V, E)$  yield a conformal system: Whenever two such paths  $P$  and  $Q$  use an arc in opposite direction, the "switched paths" (i.e., the two paths into which the vector sum  $P + Q$  decomposes after possible cycles have been removed) are suitable choices for "meet" and "join". Similarly, the collection

$$\mathcal{C} = \{(\delta^+(S), \delta^-(S)) \mid S \subseteq V\}$$

of oriented cuts in a digraph  $G = (V, E)$  yields a conformal system. Notice that the dual of the problem

$$(SF) \quad \min \left\{ u^T y \mid y(\delta^+(S)) - y(\delta^-(S)) \geq 1 \ \forall S \subseteq V, \ y \geq 0 \right\}$$

belongs to our class of generalized oriented flow problems. (SF) is a variant of a SUBMODULAR FLOW PROBLEM with constant rank function (cf. [9] or [18]). We will show below that in this special case, the oriented flow problem can be solved with the greedy algorithm.

Other examples of conformal systems are ring families as they occur in the context of bisubmodular function optimization (see, e.g., [21], or [2]): A path system  $\mathcal{P} \subseteq \{-1, 0, 1\}^{|E|}$  is called *signed ring family* if  $\mathcal{P}$  is closed w.r.t. reduced union  $\sqcup$  and intersection  $\sqcap$  on  $\{-1, 0, 1\}^{|E|}$ , defined as

$$\begin{aligned} P \sqcup Q &= ((P^+ \cup Q^+) \setminus (P^- \cup Q^-), (P^- \cup Q^-) \setminus (P^+ \cup Q^+)) \\ P \sqcap Q &= (P^+ \cap Q^+, P^- \cap Q^-). \end{aligned}$$

## 2.2 Submodularity of the Length Function

It is straightforward to check that any two non-conforming  $P, Q \in \mathcal{P}$  of a conformal system  $\mathcal{P}$  satisfy the inequality  $|(P \wedge Q)_e| + |(P \vee Q)_e| \leq |P_e| + |Q_e|$  for each  $e \in E$ . It follows that the length function of a conformal system  $\mathcal{P}$  is *submodular* in the sense

$$|P \wedge Q| + |P \vee Q| \leq |P| + |Q| \quad \text{for all non-conforming } P, Q \in \mathcal{P}.$$

In fact, a strict submodular inequality holds.

**Lemma 1** *Assume that  $\mathcal{P} \subseteq \{-1, 0, 1\}^{|E|}$  is a conformal system and  $P, Q \in \mathcal{P}$  are non-conforming. Then the following strict submodular path length inequality holds:*

$$|P \wedge Q| + |P \vee Q| < |P| + |Q|. \quad (5)$$

*Proof* Consider any oppositely oriented element  $e \in E$ . Then  $e$  lies in the support of both  $P$  and  $Q$ . On the other hand, the defining property (2) of the conformal system  $\mathcal{P}$  implies that  $e$  cannot lie in the supports of  $P \wedge Q$  and  $P \vee Q$  simultaneously. So the path length inequality must be strict.  $\square$

## 2.3 Residual Paths in Conformal Systems

Recall that the greedy algorithm raises the variable  $x_P$  to the largest currently possible value when  $P$  is the current augmenting column. So at least one capacity constraint  $\hat{e}$  becomes tight and  $P$  is only available at a later iteration when it has eventually been “unblocked” by an augmenting column  $Q$  with  $Q_{\hat{e}} \cdot P_{\hat{e}} = -1$ .

Let us call  $P' \in \mathcal{P}$  *blocked* by  $e \in E$  (w.r.t. the current  $x$ ) if either  $P'_e = 1$  and  $Ax_e = u_e$ , or  $P'_e = -1$  and  $Ax_e = l_e$ . The residual members in  $\mathcal{P}^x$  are thus those paths/columns that are not blocked by any constraint. The following (somewhat technical, but also fundamental) result tells us when and under what circumstances a path may get blocked or unblocked.

**Lemma 2** *Let  $\mathcal{P} \subseteq \{-1, 0, 1\}^{|E|}$  be a conformal system with incidence matrix  $A$  and capacities and  $l, u \in \mathbb{R}^{|E|}$ . Let  $x \in \mathbb{R}^{\mathcal{P}}$  be a feasible solution of (1) and consider two non-conforming members  $P, Q \in \mathcal{P}$ . Then the following is true:*

- (i) *If  $P, Q \in \mathcal{P}^x$ , then  $P \wedge Q \in \mathcal{P}^x$  and  $P \vee Q \in \mathcal{P}^x$  (i.e., also the residual system  $\mathcal{P}^x$  is conformal).*

- (ii) If  $P \in \mathcal{P}^x$  and  $Q$  is non-augmenting (relative to  $x$ ) but becomes augmenting after an augmentation along  $P$ , then  $P \wedge Q \in \mathcal{P}^x$  and  $P \vee Q \in \mathcal{P}^x$ .

*Proof* We first show (i) and suppose  $P \wedge Q \notin \mathcal{P}^x$ . So there exists some  $e \in E$  with either  $(P \wedge Q)_e = -1$  and  $(Ax)_e = l_e$ , or  $(P \wedge Q)_e = +1$  and  $(Ax)_e = u_e$ . Assume the former to be the case, for example. Then  $(P \vee Q)_e \leq 0$  holds (since  $P \wedge Q$  and  $P \vee Q$  are conforming). On the other hand,  $P, Q \in \mathcal{P}^x$  implies  $P_e \geq 0$  and  $Q_e \geq 0$ . So property (4) implies  $(P \wedge Q + P \vee Q)_e \geq 0$  as well, contradicting the fact that  $(P \wedge Q)_e = -1$  and  $(P \vee Q)_e \leq 0$ . A violation of the upper bound is disproved the same way.

Claim (ii) can be proved in a similar way: Assume that  $(P \wedge Q) \notin \mathcal{P}^x$  ( $P \vee Q$  can be analyzed in the same way.) So there exists some  $e \in E$  with either  $(P \wedge Q)_e = -1$  and  $(Ax)_e = l_e$ , or  $(P \wedge Q)_e = +1$  and  $(Ax)_e = u_e$ . Assume the former to be the case, for example. Again this implies  $(P \vee Q)_e \leq 0$  (since  $P \wedge Q$  and  $P \vee Q$  are conforming). Hence property (4) tells us that  $P_e + Q_e \leq -1$ . But  $P_e \geq 0$ , as  $P \in \mathcal{P}^x$ . Thus  $P_e + Q_e \leq -1$  can only happen if  $P_e = 0$  and  $Q_e = -1$ . But then  $P$  cannot unblock  $Q$ , contrary to our assumption. Hence  $P \wedge Q \in \mathcal{P}^x$  must hold indeed.  $\square$

## 2.4 Compatible Preferences

Any preference order  $(\mathcal{P}, <)$  gives rise to the definition of a *height* function  $h : \mathcal{P} \rightarrow \mathbb{Z}_+$  via

$$h(P) := \max \{k \mid \exists P_1, \dots, P_{k-1} \text{ with } P_1 < \dots < P_{k-1} < P\} \quad \forall P \in \mathcal{P}.$$

The value  $h(P)$  is usually referred to as the *height* of  $P$  and corresponds to the length of a longest chain with top element  $P$  in  $(\mathcal{P}, <)$ .

**Definition 2** A preference order  $(\mathcal{P}, <)$  on a conformal system  $\mathcal{P}$  is called *compatible* if for any two non-conformal members  $P, Q \in \mathcal{P}$  holds

$$\min\{h(P \wedge Q), h(P \vee Q)\} < \max\{h(P), h(Q)\}.$$

For the remainder of the paper, let us denote meet and join in such a way that for any two non-conformal members  $P, Q \in \mathcal{P}$  we have  $h(P \wedge Q) \leq h(P \vee Q)$ . Then  $(\mathcal{P}, <)$  is compatible if  $h(P \wedge Q) < \max\{h(P), h(Q)\}$ , i.e., if  $P \wedge Q$  is preferred to either  $P$  or  $Q$  for any two non-conforming  $P, Q \in \mathcal{P}$ .

## 2.5 Examples of Compatible Systems

**Lexicographical Ordering** Any total order on  $E$  induces an appropriate lexicographic ordering on  $\mathcal{P}$ : Write  $P < Q$  if  $Q_e \neq 0$  holds for the first element  $e \in E$  with  $|P_e - Q_e| = 1$  (i.e., the first element where the supports of  $P$  and  $Q$  disagree lies in the support of  $Q$  but not in the support of  $P$ ). Clearly, this lexicographic order yields a compatible preference relation for any conformal path system  $\mathcal{P}$ .



**Shortest Path Preference** The submodular length inequality (5) shows that shortest path length preferences

$$P \prec Q \iff |P| < |Q| \quad \forall P, Q \in \mathcal{P}$$

in conformal systems are compatible.

**Cut Lattice** The collection  $\mathcal{C} = \{(\delta^+(S), \delta^-(S)) \mid S \subseteq V\}$  of oriented cuts in a digraph  $G = (V, E)$  yields a conformal system which is compatible with respect to the ordering

$$(\delta^+(S), \delta^-(S)) \prec (\delta^+(T), \delta^-(T)) \iff S \subset T \subseteq E.$$

System  $(\mathcal{C}, \prec)$  is usually referred to as *oriented cut lattice*.

**Signed Ring Families** Recall that system  $\mathcal{P} \subseteq \{-1, 0, 1\}^{|E|}$  is a signed ring family if  $\mathcal{P}$  is closed with respect to reduced union and intersection, as defined above. It follows that  $(\mathcal{P}, \prec)$  with preference relation  $\prec$  defined via

$$P \prec Q \iff P^+ \leq Q^+ \text{ and } P^- \leq Q^- \quad \forall P, Q \in \mathcal{P}$$

is a compatible conformal system.

### 3 Running-Time on Compatible Conformal Systems

The following theorem shows that compatibility of a conformal systems guarantees the greedy algorithm to never raise the same variable twice.

**Theorem 1** *Let  $(\mathcal{P}, \prec)$  be a compatible preference relation on the conformal system  $\mathcal{P} \subseteq \{-1, 0, 1\}^{|E|}$ . Then the corresponding greedy algorithm which always augments along  $\prec$ -minimal residual paths never raises the same variable twice.*

*Proof* Let us assume that  $P$  is unblocked by augmentation along  $Q$ . Then  $P_e \cdot Q_e = -1$  for all constraints  $e \in E$  that blocked  $P$  at the time when  $Q$  was selected for augmentation. Clearly, this  $Q$  cannot be conforming with  $P$ . Hence  $P \wedge Q$  and  $P \vee Q$  exist and, since the preference order under investigation is compatible,  $P \wedge Q$  is preferred to  $P$  or  $Q$ .

After the augmentation along  $Q$ , the paths  $P$  and  $P \wedge Q$  are augmenting [cf. Lemma 2(ii)]. In case  $P \wedge Q \prec P$ , the algorithm would have chosen  $P \wedge Q$  prior to  $P$  for augmentation, and the constraint that becomes tight when  $P \wedge Q$  is raised also blocks  $P$ . So  $P \wedge Q \prec Q$  must be the case, which tells us that  $P \wedge Q$  was blocked at the moment  $Q$  was chosen.

Let  $e \in E$  be an element that blocked  $P \wedge Q$  and recall that  $P \vee Q$  is conforming with  $P \wedge Q$ . Therefore, we know (since  $\mathcal{P}$  is conformal) that  $e$  also blocks either  $P$  or  $Q$ . Since  $Q$  is actually used as an augmenting path,  $Q$  cannot be blocked. So  $P$  is blocked by  $e$ . However, this implies  $Q_e = 0$ , since  $Q$  is not blocked by  $e$ , and  $P_e + Q_e = 0$  would imply that  $e$  cannot block  $P \wedge Q$ .

The latter, however, is contradictory as it would imply that  $Q$  does *not* unblock  $P$ . We therefore conclude that a variable  $x_P$  that has been raised once during the greedy algorithm is never raised again.  $\square$

In order to bound the number of iterations of the greedy algorithm, we consider path selection rules associated to so-called *potential functions* on  $(\mathcal{P}, <)$ :

**Definition 3** A function  $g : \mathcal{P} \rightarrow \mathbb{R}$  is a *potential function* for system  $(\mathcal{P}, <)$  if

$$P < Q \implies g(P) < g(Q) \quad \forall P, Q \in \mathcal{P}.$$

We denote by  $\langle g \rangle = |\{g(P) \mid P \in \mathcal{P}\}|$  the number of different values function  $g : \mathcal{P} \rightarrow \mathbb{R}$  attains.

Note that the height function, as defined above, as well as the length function is a potential function.

The greedy algorithm on  $(\mathcal{P}, <)$  associated to potential function  $g$  always selects a residual path of minimal  $g$ -value which, by definition, is a  $<$ -minimal residual path.

The following theorem can be seen as a generalization of Edmonds–Karp’s result from ordinary max flow with shortest path preference to conformal compatible systems:

**Theorem 2** If  $(\mathcal{P}, <)$  is a compatible conformal system with potential function  $g$ , then the associated greedy algorithm which always augments along residual paths of smallest  $g$ -value terminates after at most  $\langle g \rangle |E|$  iterations.

*Proof* Let  $P_1, \dots, P_k$  denote the paths chosen for augmentation in iterations  $1, \dots, k$ . Accordingly, let  $x^{(0)}, x^{(1)}, \dots, x^{(k)}$  be the series of feasible solutions constructed during the algorithm, i.e.,  $x^{(i+1)}$  is obtained from  $x^{(i)}$  by augmentation along  $P_i$ . To shorten notation, we let  $\mathcal{P}_i = \mathcal{P}^{x^{(i)}}$  be the collection of residual paths in iteration  $i = 0, \dots, k$ . In order to prove the statement of the theorem, we show that the  $g$ -value of the augmenting paths monotonically increases throughout execution of the algorithm, and strictly increases after at most  $m = |E|$  iterations.

*Claim* For all iterations  $i = 1, \dots, k - 1$ , we have  $g(P_{i+1}) \geq g(P_i)$ . Moreover,  $g(P_{i+1}) = g(P_i)$  implies  $P_{i+1} \in \mathcal{P}_i$ .

*Proof of claim* If  $P_{i+1} \notin \mathcal{P}_i$ , then  $P_{i+1} \in \mathcal{P}_{i+1} \setminus \mathcal{P}_i$ , implying that augmentation along  $P_i$  unblocks  $P_{i+1}$ , and  $P_i$  and  $P_{i+1}$  must be non-conforming. By Lemma 2(ii), there exists some  $Q = P_i \wedge P_{i+1} \in \mathcal{P}_i$ . Since system  $(\mathcal{P}, <)$  is compatible, we know that either  $Q < P_i$  or  $Q < P_{i+1}$  must be the case. Since  $Q < P_i$  is not possible by the choice of  $P_i$ , we have  $Q < P_{i+1}$ , implying  $g(Q) < g(P_{i+1})$ . Hence,  $g(P_{i+1}) \leq g(P_i)$  would imply  $g(Q) < g(P_{i+1}) \leq g(P_i)$ , contradicting the choice of  $P_i$  (instead of  $Q$ ). Thus, indeed,  $g(P_{i+1}) > g(P_i)$  must hold.

If  $P_{i+1} \in \mathcal{P}_i$ , the inequality  $g(P_{i+1}) \geq g(P_i)$  follows immediately by the choice of  $P_i$ . Summarizing, we obtain  $g(P_{i+1}) \geq g(P_i)$  with  $g(P_{i+1}) = g(P_i)$  only if  $P_{i+1} \in \mathcal{P}_i$ .

*Claim* For any two iterations  $i, j \in [k]$  with  $j \geq i + |E|$ , we have  $g(P_j) < g(P_i)$ .

*Proof of claim* For the sake of contradiction, suppose  $g(P_j) \geq g(P_i)$ . By our claim above, it follows that  $g(P_i) = g(P_{i+1}) = \dots = g(P_j)$  and all  $|E| + 1$  paths  $P_i, P_{i+1}, \dots, P_j$  belong to  $\mathcal{P}_i$ . As a consequence of Lemma 2(i), all of these  $|E| + 1$  paths must be pairwise conforming. (Otherwise, the meet of two non-conforming paths in  $\mathcal{P}_i$  also belongs to  $\mathcal{P}_i$ , and is of smaller  $g$ -value.) However, as long as we only augment along residual paths that are pairwise conforming, none of the constraints that become tight throughout iterations  $i, i + 1, \dots, j$  can be “untightened”. Consequently, the number of iterations is bounded by  $|E|$ , the number of constraints.  $\square$

In the special case where  $\mathcal{P}$  consists only of  $\{0, 1\}$ -paths (i.e., if  $\mathcal{P} \subseteq \{0, 1\}^{|E|}$ ) every preference order  $(\mathcal{P}, <)$  is trivially a compatible, conformal system. So we may assume that the greedy algorithm was carried out with respect to the trivial preference relation with constant potential  $g = c$  and hence  $\langle g \rangle = 1$  and terminates after at most  $|E|$  iterations.

We will discuss a less trivial framework for compatible preference relations on conformal systems next.

### 3.1 Generalized Left–Right Order of Network Paths

As a further example, we show that a *generalized left–right order* on  $(s, t)$ -paths in a network (directed graph)  $G = (V, E)$  yields a compatible preference relation on a conformal system. In our discussion, we assume without loss of generality that the source  $s$  has degree 1.

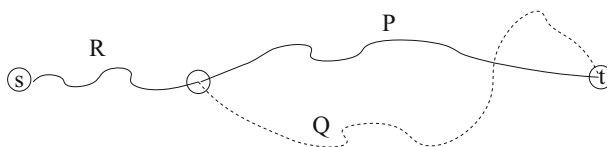
For each vertex  $v \in V$ , choose a cyclic ordering  $\pi_v$  of the edges that are incident with  $v$ . The  $\pi_v$ ’s induce an order on the set  $\mathcal{P}$  of  $(s, t)$ -paths in a natural way:

Consider two such paths  $P$  and  $Q$  and let  $R$  denote the maximal initial subpath contained in both  $P$  and  $Q$ . Then  $|R| \geq 1$  holds as we assume that  $s$  has degree 1. Let  $e$  denote the last edge in  $R$  and let  $e_P, e_Q$  be the edges following  $e$  on  $P$  resp.  $Q$ . Let  $v$  be the vertex in which  $P$  and  $Q$  split. Then we define

$$P < Q : \Longleftrightarrow \pi_v = (\dots, e, \dots, e_P, \dots, e_Q, \dots).$$

(See Fig. 1).

If  $G$  is planar embedded, for example, we can choose each  $\pi_v$  as the clockwise order of the edges around  $v$ , which induces the canonical “left to right” order on  $\mathcal{P}$ , starting with the leftmost path and ending with the rightmost path from  $s$  to  $t$ . For  $(s, t)$ -plane graphs, this left–right order  $<$  even guarantees, for non-conforming  $P, Q$ ,



**Fig. 1** Left–right order of two  $s, t$ -paths  $P < Q$

that  $P \wedge Q \prec P$  and  $Q \prec P \vee Q$ , which explains why flow is never reduced during the augmentation and non-directed paths may be disregarded completely (see [16]).

For other graphs, only compatibility can be established:

**Proposition 1** *Let  $\mathcal{P}$  be the set of  $(s, t)$ -paths of the directed graph  $G = (V, E)$ . Then the preference relation  $(\mathcal{P}, \prec)$  induced by cyclic orders  $\pi_v$  on the edges incident with the vertex  $v$  is compatible.*

*Proof* As above, we assume that  $s$  has degree 1. Consider two paths  $P$  and  $Q$  and let  $R$  denote their maximal common initial subpath. Let  $e$  denote the last edge in  $R$  and let  $e_P, e_Q$  the edges succeeding  $e$  on  $P$  resp.  $Q$ . Let  $v$  denote the vertex in which  $P$  and  $Q$  split. So  $P \prec Q$  means that  $\pi_v = (\dots, e, \dots, e_P, \dots, e_Q, \dots)$ .

Now assume that  $P$  and  $Q$  are not conforming (i.e., have a common arc that is oppositely oriented in  $P$  and  $Q$ ). Consider the (vector) sum  $F = P + Q$  (in  $\mathbb{R}^E$ ). After removing directed cycles from  $F$  (in case there are any), the resulting 2-flow decomposes into paths  $P \wedge Q$  and  $P \vee Q$  that both follow  $R$  up to the last edge  $e$  and then split into  $e_P$  resp.  $e_Q$ . So  $P \wedge Q$  (following  $e_P$ ) precedes  $Q$  in the path order.  $\square$

Recall that shortest path preference means that the greedy algorithm always chooses augmenting paths of smallest length. As mentioned earlier (cf. Sect. 2.5), this preference relation is compatible, Theorem 2 implies the Edmonds–Karp bound on the number of iterations:

**Corollary 1** *Let  $\mathcal{P} \subseteq \{-1, 0, 1\}^{|E|}$  be a conformal path system and  $(\mathcal{P}, \prec)$  the shortest length preference (i.e.,  $P \prec Q \iff |P| < |Q|$ ). Then the greedy algorithm terminates after at most  $|E|^2$  augmentations.*  $\square$

## 4 Regular Flow

Oriented matroids (see, e.g., [3] or [4]) are combinatorial abstractions of vector spaces: For  $x \in \mathbb{R}^{|N|}$  let  $X := \sigma(x) \in \{-1, 0, +1\}^{|N|}$  denote the corresponding sign vector defined by  $X^+ := \{i \in N \mid x_i > 0\}$  and  $X^- := \{i \in N \mid x_i < 0\}$ . If  $L \subseteq \mathbb{R}^{|N|}$  is a subspace, then

$$\mathcal{O} = \sigma(L) = \{\sigma(x) \mid x \in L\}$$

is the associated (linear) oriented matroid. A vector  $x \in L$  of (inclusionwise) minimal support is *elementary*. The corresponding sign vector  $X$  is referred to as a *circuit* of  $\sigma(L)$ . It is well-known (see, e.g., Prop. 5.35 in [3]) that every  $x \in L$  can be expressed as a *conformal sum*  $x = x^{(1)} + \dots + x^{(r)}$  of elementary vectors  $x^{(i)} \in L$ , i.e., a sum of pairwise conforming elementary vectors.

The oriented matroid  $\mathcal{O}$  is *regular* if  $\mathcal{O} = \sigma(L)$  with  $L = \ker U$  for some totally unimodular matrix  $U$ . In this case, elementary vectors are (up to scalar multiples) in  $\{-1, 0, +1\}^{|N|}$ . Indeed, assume, say, that the first  $k+1$  columns of  $U$  form a minimal dependent set and that

$$x_1 u_1 + \dots + x_k u_k = u_{k+1}$$

for suitable  $x_i \in \mathbb{R}$ . Then the total unimodularity of  $U$  yields  $x_i \in \{-1, 0, +1\}$  when Cramer's rule is applied.

#### 4.1 Regular Max Flow

Classical instances of regular matroids arise from the inclusion-wise minimal oriented cuts or the oriented circuits of a directed graph  $G = (V, E)$ . Consider, for example, the max flow problem (2) in  $G$  with source  $s \in V$ , sink  $t \in V$  and capacities  $u \in \mathbb{R}_+^{|E|}$ . Note that we could add an additional dummy edge  $e^* = (t, s)$  of unbounded capacity  $u^* = \infty$  without changing the max-flow problem essentially. In terms of the linear program (2), the addition of  $e^*$  means that we augment the path incidence matrix  $A$  by an all-one row  $\mathbf{1}^T$ . The  $(s, t)$ -paths  $P$  of  $G$  now correspond exactly to the circuits of the regular oriented matroid that have the entry  $+1$  in component  $e^*$ . The fact that the Ford–Fulkerson algorithm terminates with an optimal solution (if it terminates at all) generalizes from paths to "circuits minus a fixed element" in the setting of regular matroids.

Let now  $\mathcal{C}^* \in \{-1, 0, +1\}^{|N|}$  be the system of all oriented circuits  $C$  of a regular oriented matroid on  $N$  with positive coefficient  $C_{e^*} = +1$  in a fixed component  $e^* \in N$ . Set  $E = N \setminus e^*$  and let  $\mathcal{P} = \mathcal{P}(\mathcal{C}^*)$  be the induced system on  $E$  that arises from  $\mathcal{C}^*$  by restricting the vectors  $C \in \mathcal{C}^*$  to the component set  $E$ . If  $A$  is the incidence matrix of  $\mathcal{P}$ , we call problem (2), i.e.,

$$\max\{\mathbf{1}^T x \mid 0 \leq Ax \leq u\},$$

*regular max flow problem.*

The following theorem, originally due to [22], states that the greedy algorithm always returns an optimal solution of the regular max flow problem, regardless of the capacity bound  $u$ , and regardless of the preference rule for selecting residual paths for augmentation. We include a short proof for completeness.

**Theorem 3** *Let  $\hat{x}$  be a solution returned by the greedy algorithm executed on a regular max flow problem. Then  $\hat{x}$  is an optimal solution.*

*Proof* Augment the matrix  $A$  to the incidence matrix  $A^*$  of  $\mathcal{C}^*$  by adding the row  $\mathbf{1}^T = (1, \dots, 1)$  with all coefficients 1. Then we want to maximize  $\mathbf{1}^T x$  under the constraints

$$0 \leq A^* x = \begin{bmatrix} A \\ \mathbf{1}^T \end{bmatrix} x \leq \begin{bmatrix} u \\ \infty \end{bmatrix}.$$

Suppose to the contrary that the statement of the Theorem is false and there is an improving "ascent direction"  $d$  such that  $x = \hat{x} + d$  yields a better feasible solution. Then the vector  $y = A^* d$  must satisfy  $y_{e^*} = \mathbf{1}^T d > 0$ . The vector  $y$ , being a member of the column space of  $A^*$ , is in  $\ker U$ . Hence  $y$  is a conormal sum of oriented circuits. In particular, there must be one circuit  $C$  with  $C_{e^*} = 1$  that induces an improving augmenting column  $P \in \mathcal{P}$ , which contradicts the fact that the greedy algorithm has come to a halt with  $\hat{x}$ .  $\square$

It is easy to see that the system  $\mathcal{P} = \mathcal{P}(\mathcal{C}^*)$ , as defined above, is conformal: Assume that  $P$  and  $Q$  are non-conforming. Then the sum  $P + Q$  is integral and, extended by a “+2” entry in component  $e^*$ , yields a vector  $s$  in the kernel of the totally unimodular matrix  $U$ . This vector  $s$  can be written as conformational sum of elementary  $(-1, 0, +1)$ -vectors, two of them with a “+1” entry in coordinate  $e^*$ . These two vectors, with coordinate  $e^*$  removed, may be taken as  $P \wedge Q$  and  $P \vee Q$ .

In view of Theorem 2, we therefore conclude

**Corollary 2** *If  $\mathcal{P}$  satisfies the hypothesis of Theorem 3 and  $(\mathcal{P}, <)$  is a compatible preference relation, then the greedy algorithm never steps backwards and terminates with an optimal solution. In particular, the shortest path rule ensures that an optimal solution for the regular max flow problem is attained after at most  $|E|^2$  path augmentations.*  $\square$

*Remark* Regular max flow is already studied by Minty [27] (see also [24]). In particular, it seems that Bland [5] reported on the generalization of the Edmonds–Karp rule to regular flows. Unfortunately, there is no written contribution of Bland’s result in the corresponding conference proceedings. Indeed, the result seems to have not been published independently.

## 4.2 Cycle Cancellation

Our analysis may also be applied to problems with more general objective functions, e.g., min cost flow on directed graphs. In the latter case, the well-known “cycle cancelling” method can be seen as a greedy algorithm seeking for a min cost circulation in a weighted, capacitated digraph. The latter has actually been studied in detail before, even in the context of regular (or “unimodular”) spaces (cf. [7]) and with convex (rather than linear) cost functions (cf. [25]). We therefore comment only briefly on this topic (in the maximization setting that we have used throughout in our paper).

Let  $\mathcal{C} \subseteq \{-1, 0, +1\}^{|E|}$  be the family of oriented circuits of a regular oriented matroid on  $E$ . Assume that  $w \in \mathbb{R}^E$  is a vector of given weights and set  $w(P) = w(P^+) - w(P^-)$  for all  $P \in \mathcal{C}$ . We seek to solve the linear program

$$\max \left\{ \sum_{P \in \mathcal{C}} w(P)x_P \mid l \leq \sum_{P \in \mathcal{C}} x_P P \leq u \right\}.$$

Given two non-conforming circuits  $P, Q \in \mathcal{C}$  (interpreted as vectors in  $\mathbb{R}^{|E|}$ ), their sum can be expressed as a conformational sum of circuits  $R_i \in \mathcal{C}$ ,

$$P + Q = R_1 + \cdots + R_t.$$

If  $t = 1$ , we set  $P \wedge Q = R_1$  and  $P \vee Q = 0$ . If  $t \geq 2$ , we seek to choose  $P \wedge Q$  and  $P \vee Q$  such that, in a suitable preference relation on  $\mathcal{C} \cup \{0\}$ , at least one of them precedes  $P$  or  $Q$ . A well-known preference relation achieving this is the so-called *mean value* relation with potential  $-m$ ,

$$P \prec Q \iff m(P) > m(Q), \quad \text{with } m(P) = \frac{w(P)}{|P|}.$$

The mean value preference which selects paths with highest mean is compatible. Indeed, assume, say,  $m(P) \geq m(Q)$ . In the first case  $P + Q \in \mathcal{C}$ , we then find

$$\frac{w(P \wedge Q)}{|P \wedge Q|} = \frac{w(P + Q)}{|P + Q|} \geq \frac{w(P) + w(Q)}{|P| + |Q| - 2} > \frac{w(P) + w(Q)}{|P| + |Q|} \geq \frac{w(Q)}{|Q|}$$

as required. In the second case, where  $P + Q = R_1 + \dots + R_t$  is the conformal sum of  $t \geq 2$  circuits, we choose for  $P \wedge Q$  and  $P \vee Q$  the two circuits with highest mean value to ensure that, say,

$$\frac{w(P \wedge Q)}{|P \wedge Q|} = \max_i \frac{w(R_i)}{|R_i|} \geq \frac{\sum_i w(R_i)}{\sum_i |R_i|} \geq \frac{w(P) + w(Q)}{|P| + |Q| - 2} > \frac{w(Q)}{|Q|}$$

as before. Hence  $P \wedge Q$  is preferred to  $Q$  in the mean value order. For later use we note that the inequalities above can actually be strengthened to

$$\frac{w(P \wedge Q)}{|P \wedge Q|} \geq \frac{|P| + |Q|}{|P| + |Q| - 2} \frac{w(Q)}{|Q|} \geq \frac{|E|}{|E| - 1} \frac{w(Q)}{|Q|} \quad (6)$$

Theorem 2 now implies that the greedy algorithm produces an optimal solution after at most  $\langle m \rangle |E|$  iterations. As a consequence, the solution can be found in polynomial time (even a strongly polynomial time bound can be shown (cf. [25])). Most polynomial time proofs for the mean cycle method are based on dual arguments. So it might be interesting to observe that a "primal" proof is quite straightforward in our framework along the lines of the proof of Theorem 2.

Assume that the greedy algorithm increases  $x$  on the circuits  $P_1, \dots, P_t$  in this order. Each augmentation on  $P = P_i$  tightens some constraint  $e \in E$  (an upper bound in case  $e \in P^+$  and a lower bound in case  $e \in P^-$ ). As a consequence, any  $|E|$  consecutive augmentations must contain a non-conforming pair  $P_i, P_j$ ,  $i < j$ . We claim that

$$m(P_j) = \frac{w(P_j)}{|P_j|} \leq \frac{w(P_i)}{|P_i|} \frac{|E| - 1}{|E|} = m(P_i) \frac{|E| - 1}{|E|}$$

holds for any such pair, showing that after at most  $|E|$  augmentations, the mean value has decreased by at least a factor of  $\frac{|E|-1}{|E|}$ . The latter implies that after at most  $|E|^2$  augmentations, the mean value has decreased at least by a factor 1/2. As the mean value ranges from at most  $w_{\max}$ , the maximum arc weight, to  $1/|E|$  (assuming integer weights), we find that the number of augmentations is of order

$$O(|E|^2 \log(|E| w_{\max})).$$

The desired decrease follows readily from our observations above: Assume, say, that  $P = P_i$  and  $Q = P_j$  are the two non-conforming circuits. We may assume as

well that all  $P_k$  with  $i < k < j$  are conforming with  $Q$  as otherwise we can replace  $i$  with  $k$  and conclude

$$\frac{w(P_i)}{|P_i|} \geq \frac{w(P_k)}{|P_k|} \geq \frac{w(P_j)}{|P_j|} \frac{|E|}{|E| - 1}.$$

So  $Q = P_j$  is eligible for the greedy algorithm immediately after  $P = P_i$  has been augmented. Hence  $P \wedge Q$  must be eligible already at the time when  $P = P_i$  gets augmented. The only reason for selecting  $P$  instead of  $P \wedge Q$  thus must be that  $m(P) \geq m(P \wedge Q)$  is true. Hence indeed we may conclude from (6) that

$$m(P) = \frac{w(P)}{|P|} \geq \frac{w(Q)}{|Q|} \frac{|E|}{|E| - 1} = \frac{|E|}{|E| - 1} m(Q)$$

follows, proving our claim.

## 5 Open Problems

One of the most interesting questions in our opinion concerns the generalized left–right order of network paths in the classical setting: Is there a polynomial bound on the number of augmentations when ordinary paths are ordered by the generalized left–right order, i.e., if we use depth first search (according to suitably chosen cyclic orderings  $\pi_v$ ) for constructing the augmenting paths, instead of breadth first search as usual? Recent work by Dean et al. [8] implies that some cyclic orderings require exponential time—but maybe others (possibly randomly chosen ones) work well?

**Acknowledgements** We are grateful to Jon Lee [26] for pointing our attention to Bland [5] and to an anonymous referee for making us aware of several other related publications.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Englewood Cliffs (1993)
2. Ando, K., Fujishige, S.: On structures of bisubmodular polyhedra. Math. Program. **74**, 293–317 (1996)
3. Bachem, A., Kern, W.: Linear Programming Duality: An Introduction to Oriented Matroids. Springer, Berlin (1992)
4. Björner, A., Las Vergnas, M., Sturmfels, B., White, N., Ziegler, G.: Oriented Matroids. Cambridge University Press, Cambridge (1993)
5. Bland, R.: Fast algorithms for totally unimodular linear programming. In: Lecture at XIth International Symposium on Mathematical Programming, Bonn (1982)
6. Borradaile, G., Klein, P.: An  $O(n \log n)$  algorithm for maximum  $st$ -flow in a directed planar graph. In: SODA06 Proceedings, pp. 524–533 (2006)
7. Burkard, R.E., Hamacher, H.: Minimal cost flows in regular matroids. Math. Program. Study **14**, 32–47 (1981)



8. Dean, B.C., Goemans, M.X., Immorlica, N.: Finite Termination of Augmenting Path Algorithms in the Presence of Irrational Problem Data. ESA 2006. Lecture Notes in Computer Science, vol. 4168. Springer, Berlin, Heidelberg
9. Edmonds, J., Giles, R.: A min–max relation for submodular functions on graphs. *Ann. Discrete Math.* **1**, 185–204 (1977)
10. Edmonds, J., Karp, R.M.: Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM* **19**(2), 248–264 (1972)
11. Faigle, U., Hoffman, A.J., Kern, W.: A characterization of non-negative greedy matrices. *SIAM J. Discrete Math.* **9**, 1–6 (1996)
12. Faigle, U., Kern, W., Still, G.: *Algorithmic Principles of Mathematical Programming*. Kluwer Texts in Mathematical Sciences. Kluwer Academic Publisher, Dordrecht (2002)
13. Faigle, U., Peis, B.: Two-phase greedy algorithm for some classes of combinatorial linear programs. In: SODA08 Proceedings (2008)
14. Faigle, U., Kern, W., Peis, B.: A ranking model for cooperative games, convexity and the greedy algorithm. *Math. Program.* **132**, 303–407 (2012)
15. Faigle, U., Kern, W., Peis, B.: On greedy and submodular matrices. In: Proceedings of TAPAS 2011, Springer LNCS, p. 6595 (2011)
16. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. *Can. J. Math.* **8**, 399–404 (1956)
17. Frank, A.: Increasing the rooted connectivity of a digraph by one. *Math. Program.* **84**, 565–576 (1999)
18. Frank, A.: Submodular flows. In: Frank, A. (ed.) *Connections in Combinatorial Optimization*. Oxford Lecture Series in Mathematics and its Applications, vol. 38. Oxford University Press, Oxford (2011)
19. Fujishige, S.: Principal structures in submodular systems. *Discrete Appl. Math.* **2**, 77–79 (1980)
20. Fujishige, S.: *Submodular Functions and Optimization*. Annals of Discrete Mathematics, vol. 58, 2nd edn. Elsevier, Amsterdam (2005)
21. Fujishige, S., Iwata, S.: Bisubmodular function minimization. *SIAM J. Discrete Math.* **19**, 1065–1073 (2006)
22. Hamacher, H.: Algebraic flows in regular matroids. *Discrete Appl. Math.* **2**, 27–38 (1980)
23. Hoffman, A.J., Kolen, A.W.J., Sakarovitch, M.: Totally balanced and greedy matrices. *SIAM J. Algebraic Discrete Methods* **6**, 721–730 (1985)
24. Hochstaettler, W., Nickel, R.: Note on a MaxFlowMinCut property for oriented matroids. Technical Report feU-dmo008.07 (2007). <http://www.fernuni-hagen.de/MATHEMATIK/DMO/pubs/feu-dmo008-07>
25. Karzanov, A.V., McCormick, S.T.: Polynomial methods for separable convex optimization in unimodular linear spaces with applications. *SIAM J. Comput.* **26**(4), 1245–1275 (1997)
26. Lee, J.: Personal communication (2012)
27. Minty, G.J.: On the axiomatic foundations of the theories of directed linear graphs, electrical networks and network programming. *J. Math. Mech.* **15**, 485–520 (1966)